What is claimed is:

1. A method for detecting device-enhanced memory modules in a system including one or more memory modules comprising system memory, wherein at least one of the one or more memory modules is a device-enhanced memory module comprising one or more embedded devices, the method comprising:

writing a data sequence to a first memory module of the one or more memory modules in the system;

reading the data sequence from the first memory module; and

identifying the first memory module as a device-enhanced memory module if the data sequence as read from the first memory module is modified from the data sequence as written to the first memory module.

2. The method of claim 1, further comprising:

identifying the first memory module as not being a device-enhanced memory module if the data sequence read from the first memory module is the same as the data sequence as written to the first memory module.

3. The method of claim 1, wherein the data sequence is modified by an embedded device comprised on the first memory module prior to said reading the data sequence from the first memory module.

4. The method of claim 1, wherein the data sequence includes one or more computer instructions that are illegal instructions for reads and writes to the system memory during normal program execution within the system, wherein the one or more illegal computer instructions are operable to prevent the data sequence from being accidentally written to the system memory during normal operation of the system.

5. The method of claim 1, further comprising, after said identifying the first memory module as a device-enhanced memory module:

writing a plurality of data writes to the first memory module; and

the first memory module learning an error detection and correction scheme of the system from the plurality of data writes.

6. The method of claim 5, wherein each of the plurality of data writes includes a plurality of data bits and a plurality of check bits, wherein the plurality of data writes includes one data write for each of the plurality of data bits, wherein the particular data bit is set in the data write and all other data bits are not set, and wherein the check bits are set as appropriate for the data write in the error detection and correction scheme.

7. The method of claim 6, wherein said learning the error detection and correction scheme comprises:

generating check bit masks for each of the plurality of check bits using the check bits and the data bits of each of the data writes with one data bit set; and

storing the generated check bit masks;

wherein, when applying the learned error detection and correction scheme, the check bit masks are used in generating check bits for data writes to and from the first memory module.

8. The method of claim 6, wherein said learning the error detection and correction scheme comprises:

for each of the data writes with one set data bit:

storing a position of the set data bit in the plurality of data bits in a syndrome array at a location indicated by the check bits of the data write;

wherein the check bits of each of the data writes for the plurality of data bits define an error detection and correction code associated with the data bit position of the set data bit in the data write;

wherein, when applying the learned error detection and correction scheme, each of the stored data bit positions is retrievable from the syndrome array using the error detection and correction code associated with the particular data bit position.

9. The method of claim 6, wherein the plurality of data writes further includes one data write with no data bits set, wherein the check bits of the data write are set as appropriate for the data write with no data bits set in the error detection and correction scheme.

10. The method of claim 9, further comprising:

storing the check bits for the data write with no data bits set;

wherein, when applying the learned error detection and correction scheme, the stored check bits for the data write with no data bits set are used as a check bit inversion mask for data writes to and from the first memory module.

11. The method of claim 5, further comprising:

configuring error detection and correction logic of the first memory module to use the learned error detection and correction scheme;

wherein the configured error detection and correction logic is operable to provide error detection and correction for data writes to and from the first memory module using the learned error detection and correction scheme.

12. The method of claim 5, further comprising verifying the learned error detection and correction scheme for the first memory module after said learning the error detection and correction scheme.

13. The method of claim 5, further comprising:

writing a plurality of data values to the first memory module;

the first memory module generating error detection and correction data for the plurality of data values using the learned error detection and correction scheme;

examining the generated error detection and correction data; and

if the generated error detection and correction data is correct according to the learned error detection and correction scheme, indicating that said learning the error detection and correction scheme succeeded; and

if the generated error detection and correction data is not correct according to the learned error detection and correction scheme, indicating that said learning the error detection and correction scheme failed.

14. The method of claim 1, further comprising identifying interleave settings for the first memory module after said identifying the first memory module as a device-enhanced memory module, wherein the interleave settings include an interleave value and a number of cache lines used for interleaving the system memory.

15. The method of claim 14, further comprising verifying the identified interleave settings for the first memory module.

16. The method of claim 15, wherein said verifying the identified interleave settings comprises:

writing a first block of data to an input buffer on the first memory module, wherein the input buffer is used by an embedded device on the first memory module for receiving input data;

the embedded device reading the first block of data from the input buffer;

the embedded device writing the first block of data to an output buffer on the first memory module, wherein the output buffer is used by the embedded device for outputting data;

reading the first block of data from the output buffer;

examining the first block of data read from the output buffer; and

verifying the interleave settings in response to said examining the first block of data read from the output buffer.

17. The method of claim 1, further comprising:

writing a plurality of data values to each of a plurality of burst lines in a memory region of the first memory module, wherein, if the first memory module is interleaved with one or more other memory modules, a first portion of the plurality of data values are written to the memory region of the first memory module and a second portion of the plurality of data values are written to the one or more other memory modules with which the first memory module is interleaved;

reading one or more of the plurality of data values written to the memory region; and

examining the one or more data values read from the memory region to determine the interleave characteristics of the first memory module.

18. The method of claim 1, further comprising, after said identifying the first memory module as a device-enhanced memory module:

reading one or more hardware identification values from the first memory module; and

storing the one or more hardware identification values in a memory region accessible by a device driver for the first memory module.

19. The method of claim 1, further comprising, after said identifying the first memory module as a device-enhanced memory module:

writing one or more configuration values to one or more of a plurality of registers on the first memory module, wherein the configuration values are configured for use during operation of an embedded device on the first memory module;

wherein the configuration values include one or more of a sleep mode timer value, a restart timer value, an error timer value, and a driver version of a device driver for the embedded device on the first memory module.

20. The method of claim 1, wherein said writing, said reading, and said identifying are performed during system startup.

21. The method of claim 1, further comprising, after said identifying the first memory module as a device-enhanced memory module:

learning an error detection and correction scheme for the first memory module;

identifying interleave settings for the first memory module;

performing a buffer check for the first memory module; and

storing one or more hardware identification values to one or more registers on the first memory module.

22. The method of claim 21, wherein said writing, said reading, said identifying, said learning, said identifying, said performing, and said storing are performed during system startup.

23. The method of claim 1, further comprising, after said identifying the first memory module as a device-enhanced memory module, wherein said initializing includes:

initializing the first memory module to use an error detection and correction scheme of the system; and

initializing the first memory module to use a memory interleave method, if any, of the system;

wherein, after said initializing the first memory module to use an error detection and correction scheme and said initializing the first memory module to use a memory interleave method, an embedded device on the first memory module is accessible to perform one or more functions on the system.

24. The method of claim 1, further comprising:

initializing the first memory module to operate in a system memory implementation after said identifying the first memory module as a device-enhanced memory module; and

sending an end initialization sequence to the first memory module after said initializing;

wherein, after receiving the end initialization sequence, an embedded device on the first memory module is accessible to perform one or more functions on the system.

25. The method of claim 1, further comprising:

initializing the first memory module to operate in a system memory implementation after said identifying the first memory module as a device-enhanced memory module; and

wherein, after said initializing, an embedded device on the first memory module is accessible to perform one or more functions on the system.

26. The method of claim 25, further comprising:

the embedded device on the first memory module entering sleep mode after said initializing, wherein, in the sleep mode, the embedded device is not accessible to perform the one or more functions, and wherein the embedded device operates with reduced power requirements in sleep mode.

27. The method of claim 26, further comprising, after said entering sleep mode, repeating said writing, said reading, said identifying and said initializing, wherein, after said repeating, the embedded device on the first memory module is again accessible to perform the one or more functions on the system.

28. The method of claim 1, wherein an embedded device on the first memory module is a compression/decompression engine.

29. A method for detecting device-enhanced memory modules in a system including a plurality memory modules comprising system memory, wherein one or more of the plurality of memory modules are device-enhanced memory modules each comprising one or more embedded devices, the method comprising:

writing one or more data sequences including a last data sequence to each of a plurality of memory regions in the system memory;

reading at least the last data sequence from each of the plurality of memory regions in the system memory; and

identifying one or more of the plurality of memory modules as device-enhanced memory modules in response to said reading the at least the last data sequence from each of the plurality of memory regions in the system memory;

wherein a first memory module of the plurality of memory modules is identified as a device-enhanced memory module if the at least the last data sequence read from a memory region on the first memory module is different than the at least the last data sequence as written to the memory region on the first memory module.

30. The method of claim 29, wherein a second memory module is identified as not being a device-enhanced memory module if the at least the last data sequence read from a memory region on the second memory module is the same as the at least the last data sequence as written to the memory region on the second memory module.

31. The method of claim 29, wherein the plurality of memory regions includes at least one memory region on each of the plurality of memory modules comprising the system memory.

32. The method of claim 29, wherein the data sequence is modified by an embedded device comprised on each of the one or more memory modules identified as device-enhanced memory modules prior to said reading the at least the last data sequence from the one or more memory modules identified as device-enhanced memory modules.

33. The method of claim 29, further comprising, after said identifying the one or more memory modules as device-enhanced memory modules:

writing a plurality of data writes to each of the one or more memory modules identified as device-enhanced memory modules;

each of the one or more memory modules identified as device-enhanced memory modules learning an error detection and correction scheme of the system from the plurality of data writes; and

configuring error detection and correction logic of each of the one or more memory modules identified as device-enhanced memory modules in accordance with the learned error detection and correction scheme;

wherein the configured error detection and correction logic of each of the one or more memory modules identified as device-enhanced memory modules is operable to provide error detection and correction for data writes to and from the memory module in accordance with the learned error detection and correction scheme.

34. The method of claim 29, further comprising:

identifying interleave settings for each of the one or more memory modules identified as device-enhanced memory modules, wherein the interleave settings include an interleave value and a number of cache lines used for interleaving the system memory.

35. The method of claim 29, further comprising:

reading one or more hardware identification values from each of the one or more memory modules identified as device-enhanced memory modules; and

storing the one or more hardware identification values in one or more memory regions accessible by one or more device drivers for the one or more memory modules identified as device-enhanced memory modules.

36. The method of claim 29, further comprising:

for each of the one or more memory modules identified as device-enhanced memory modules:

writing one or more configuration values to one or more of a plurality of registers on the memory module, wherein the configuration values are configured for use during operation of an embedded device on the memory module;

wherein the configuration values include one or more of a sleep mode timer value, a restart timer value, an error timer value, and a driver version of a device driver for the embedded device on the memory module.

37. The method of claim 29, wherein said writing, said reading, and said identifying are performed during system startup.

38. The method of claim 29, further comprising:

initializing the one or more memory modules to operate in a system memory implementation after said identifying the one or more memory modules as device-enhanced memory modules; and

wherein, after said initializing, each of one or more embedded devices on each of the one or more memory modules is accessible to perform one or more functions on the system.

39. The method of claim 29, wherein one or more embedded devices on the one or more memory modules identified as device-enhanced memory modules are compression/decompression engines.

40. A method for learning an error detection and correction scheme in a system including one or more memory modules comprising system memory, the method comprising:

writing a plurality of data writes to a first memory module of the one or more memory modules, wherein the first memory module is a device-enhanced memory module comprising one or more embedded devices;

learning the error detection and correction scheme of the system from the plurality of data writes; and

configuring error detection and correction logic of the first memory module in accordance with said learned error detection and correction scheme;

wherein the configured error detection and correction logic of the first memory module is operable to provide error detection and correction for data writes to and from the first memory module using the learned error detection and correction scheme.

41. The method of claim 40, wherein the system further includes a memory bus comprising a plurality of memory bus data bits and one or more check bits, wherein the memory bus is coupled to the one or more memory modules, wherein each data write of the plurality of data writes includes a data portion comprising one data bit for each memory bus data bit and a check portion comprising one check bit for each memory bus check bit;

wherein, for each data bit in the data portion, the plurality of data writes includes one data write with the particular data bit set in the data portion of the data write and all other data bits in the data portion not set, and wherein zero or more of the check bits in the check portion of the data write are set to indicate an error detection and correction code of the error detection and correction scheme to be learned for the set data bit of the data write.

42. The method of claim 41, wherein the first memory module learning the error detection and correction scheme comprises:

for each of the plurality of data writes with one data bit set, storing the position of the set data bit in the data portion at a location determined by the error detection and correction code of the data write;

wherein, in said applying the learned error detection and correction scheme, each stored data bit position is retrievable using the error detection and correction code associated with the particular data bit position, and wherein the stored data bit positions retrieved using the error detection and correction codes associated with the data bit positions are used to determine which, if any, of the data bits in the received data are in error.

43. The method of claim 41, wherein the first memory module learning the error detection and correction scheme comprises:

generating check bit masks for each of the check bits in the check portion using the check portion and the data portion of each of the plurality of data writes for the plurality of data bits;

wherein the check bit masks are configured for use in generating check bits for the data received by and transmitted from the first memory module when applying the learned error detection and correction scheme.

44. The method of claim 41, wherein each of the check bits is associated with a check bit mask, the method further comprising:

for each of the zero or more set check bits of the check portion of each of the plurality of data writes:

performing a logical OR operation on the contents of the check bit mask associated with the check bit and the data portion of the data write; and

storing the results of the OR operation in the check bit mask associated with the check bit;

wherein the check bit masks are configured for use in generating the check bits for the data received by and transmitted from the first memory module when applying the learned error detection and correction scheme.

45. The method of claim 40, wherein the plurality of data writes includes one data write with no data bits of the data portion set, and wherein zero or more of the check bits of the data write are set to indicate an error detection and correction code of the error detection and correction scheme to be learned for the data portion with no data bits set, the method further comprising:

storing the error detection and correction code for the data write;

wherein the stored error detection and correction code for the data write is configured for use as a check bit inversion mask when applying the learned error detection and correction scheme.

46. The method of claim 40, wherein said learning the error detection and correction scheme comprises:

learning a check bit mask for each of one or more check bits used in the error detection and correction scheme; and

learning a check bit inversion mask for the error detection and correction scheme.

47. The method of claim 40, further comprising:

learning a check bit mask for each of one or more check bits used in the error detection and correction scheme.

applying the learned error detection and correction scheme to a first data write received on the first memory module, wherein the first data write comprises a plurality of data bits;

wherein said applying the learned error detection and correction scheme to the first data write comprises using each of the learned check bit masks to select one or more of the data bits from the first data write for use in generating the check bit associated with the learned check bit mask.

48. The method of claim 40, further comprising:

learning a check bit mask for each of one or more check bits used in the error detection and correction scheme.

applying the learned error detection and correction scheme to a first data write received on the first memory module, wherein the first data write comprises a plurality of data bits;

wherein said applying the learned error detection and correction scheme to the first data write comprises, for each of the learned check bit masks:

performing a logical AND operation on the check bit mask and the data bits from the first data write; and

performing a logical Exclusive OR (XOR) operation on the output of the logical AND operation, wherein the output of the logical XOR operation is the check bit associated with the check bit mask;

wherein the generated check bits are used to verify the correctness of the data bits from the first data write.

49. The method of claim 48, wherein said applying the learned error detection and correction scheme further comprises:

learning a check bit inversion mask for the error detection and correction scheme;

performing a logical XOR operation on each generated check bit and the learned check bit inversion mask to generate a system check bit; and

concatenating each of the generated system check bits with the data bits of the first data write in preparation for transmitting the first data write from the first memory module.

50. The method of claim 40, wherein said learning the error detection and correction scheme further comprises:

learning which one of a plurality of error detection and correction codes is associated with each of the memory bus data bit positions; and

storing information on the learned associations in a syndrome array;

wherein, in said applying the learned error detection and correction scheme, each data bit position is retrievable from the syndrome array using the error detection and correction code associated with the particular data bit position.

51. The method of claim 50, further comprising:

applying the learned error detection and correction scheme to a first data write received on the first memory module, wherein said applying the learned error detection and correction scheme to the first data write comprises:

generating check bits for the first data write using the learned error detection and correction scheme;

detecting one or more data bit errors, if any, in the first data write received by the first memory module using the generated check bits; and

if the detected one or more data bit errors are correctible using the learned error detection and correction scheme, correcting the one or more detected data bit errors using the syndrome array to locate the one or more data bits of the first data write that are in error.

52. The method of claim 40, further comprising:

applying the learned error detection and correction scheme to a first data write received on the first memory module, wherein the first data write includes a plurality of data

bits and one or more check bits, wherein said applying the learned error detection and correction scheme to the first data write comprises:

generating check bits for the first data write using the learned error detection and correction scheme;

performing a logical Exclusive OR (XOR) operation on the one or more check bits received in the first data write data write and a check bit inversion mask of the learned error detection and correction scheme;

comparing the output of the logical XOR operation with the generated check bits; and

if the output of the logical XOR operation does not match the generated check bits, correcting one or more of the data bits received in the data portion of the first data write; and

if the output of the logical XOR operation matches the generated check bits, outputting the data bits received as a data portion of the first data write as correct output data.

53. The method of claim 52, wherein said correcting the one or more of the data bits comprises:

locating the bit positions of the one or more data bits to be corrected in a syndrome array of the learned error detection and correction scheme; and

inverting the data bits at the located bit positions in the data portion of the first data write.

54. The method of claim 40, wherein at least one of the one or more embedded devices on the first memory module is a compression/decompression engine.

55. A method for learning an error detection and correction scheme in a system including system memory, a memory bus comprising a plurality of memory bus data bits

and one or more memory bus check bits, and a plurality of memory masters operable to send and receive data on the memory bus, the method comprising:

a first memory master receiving a plurality of data writes from a second memory master; and

the first memory master learning an error detection and correction scheme of the second memory master from the received plurality of data writes;

wherein, after said learning the error detection and correction scheme, the first memory master is operable to apply the learned error detection and correction scheme to data received by and transmitted from the first memory master.

56. The method of claim 55, wherein each data write of the plurality of data writes includes a data portion comprising one data bit for each memory bus data bit and a check portion comprising one check bit for each memory bus check bit;

wherein, for each memory bus data bit, the plurality of data writes includes one data write with the particular data bit set in the data portion of the data write and all other data bits in the data portion not set, and wherein zero or more of the check bits in the check portion of the data write are set to indicate an error detection and correction code of the error detection and correction scheme to be learned for the particular data bit that is set the data write.

57. The method of claim 56, wherein the first memory master learning the error detection and correction scheme comprises:

for each of the plurality of data writes with one data bit set, storing the position of the set data bit in the data portion at a location determined by the error detection and correction code of the data write;

wherein, in said applying the learned error detection and correction scheme, each stored data bit position is retrievable using the error detection and correction code associated with the particular data bit position, and wherein the stored data bit positions retrieved using the error detection and correction codes associated with the data bit positions are used to determine which, if any, of the data bits in the received data are in error.

58. The method of claim 56, wherein the first memory master learning the error detection and correction scheme comprises:

generating check bit masks for each of the check bits in the check portion using the check portion and the data portion of each of the plurality of data writes for the plurality of data bits;

wherein the check bit masks are configured for use in generating check bits for the data received by and transmitted from the first memory master when applying the learned error detection and correction scheme.

59. The method of claim 56, wherein the plurality of data writes includes one data write with no data bits of the data portion set, and wherein zero or more of the check bits for the data write are set to indicate an error detection and correction code of the error detection and correction scheme to be learned for the data portion with no data bits set, the method further comprising:

storing the error detection and correction code for the data write;

wherein the stored error detection and correction code for the data write is configured for use as a check bit inversion mask when applying the learned error detection and correction scheme.

60. The method of claim 55, further comprising:

applying the learned error detection and correction scheme to a first data write received on the first memory master, wherein the first data write comprises a plurality of data bits;

wherein said applying the learned error detection and correction scheme to the first data write comprises:

generating check bits for the first data write using the learned error detection and correction scheme;

generating a system check bit from each generated check bit; and

concatenating each of the generated system check bits with data bits of the first data write for sending to another memory master.

61. The method of claim 55, wherein said learning the error detection and correction scheme further comprises:

learning which one of a plurality of error detection and correction codes is associated with each of the memory bus data bit positions; and

storing information on the learned association in a syndrome array;

wherein, in said applying the learned error detection and correction scheme, each data bit position is retrievable from the syndrome array using the error detection and correction code associated with the particular data bit position.

62. The method of claim 61, further comprising:

applying the learned error detection and correction scheme to a first data write received on the first memory master, wherein the first data write includes a plurality of data bits, wherein said applying the learned error detection and correction scheme to the first data write comprises:

generating check bits for the first data write received by the first memory master using the learned error detection and correction scheme;

detecting one or more data bit errors, if any, in the first data write received by the first memory master using the generated check bits; and

if the detected one or more data bit errors are correctible using the learned error detection and correction scheme, correcting the one or more detected data bit errors using the syndrome array to locate the one or more data bits that are in error.

63. The method of claim 55, wherein the first memory master is a device-enhanced memory module comprising one or more embedded devices.

64. The method of claim 63, wherein at least one of the one or more embedded devices on the first memory master is a compression/decompression engine.

65. A system comprising:

one or more memory modules comprising system memory, wherein at least one of the one or more memory modules comprises one or more embedded devices;

a device driver executable within the system to:

write a data sequence to a first memory module of the one or more memory modules;

read the data sequence from the first memory module; and

identify the first memory module as a device-enhanced memory module if the data sequence as read from the first memory module is modified from the data sequence as written to the first memory module.

66. The system of claim 65, wherein the device driver is further executable to:

identify the first memory module as not being a device-enhanced memory module if the data sequence read from the first memory module is the same as the data sequence as written to the first memory module.

67. The system of claim 65, wherein the first memory module is configured to modify the data sequence prior to said reading the data sequence from the first memory module.

68. The system of claim 65, wherein, after said identifying the first memory module as a device-enhanced memory module, the device driver is further executable to:

write a plurality of data writes to the first memory module; and

wherein the first memory module is configured to:

learn an error detection and correction scheme of the system from the plurality of data writes; and

apply the learned error detection and correction scheme to data received by and transmitted from the first memory module.

69. The system of claim 68, wherein each of the plurality of data writes includes a plurality of data bits and a plurality of check bits, wherein the plurality of data writes includes one data write for each of the plurality of data bits, wherein the particular data bit is set in the data write and all other data bits are not set, and wherein the check bits are set as appropriate for the data write in the error detection and correction scheme.

70. The system of claim 69, wherein, in said learning the error detection and correction scheme, the first memory module is further configured to:

generate check bit masks for each of the plurality of check bits using the check bits and the data bits of each of the data writes with one data bit set; and

store the generated check bit masks;

wherein, in said applying the learned error detection and correction scheme, the first memory module is further configured to generate check bits for data writes to and from the first memory module using the check bit masks.

71. The system of claim 69, wherein, in said learning the error detection and correction scheme, the first memory module is further configured to:

for each of the data writes with one data bit set:

store a position of the set data bit in the plurality of data bits in a syndrome array at a location indicated by the check bits of the data write;

wherein the check bits of each of the data writes for the plurality of data bits define an error detection and correction code associated with the data bit position of the set data bit in the data write;

wherein, in said applying the learned error detection and correction scheme, the first memory module is further configured to retrieve each of the stored data bit positions from the syndrome array using the error detection and correction code associated with the particular data bit position.

72. The system of claim 69, wherein the plurality of data writes further includes one data write with no data bits set, wherein the check bits of the data write are set as appropriate for the data write with no data bits set in the error detection and correction scheme, and wherein, in said learning the error detection and correction scheme, the first memory module is further configured to:

store the check bits for the data write with no data bits set;

wherein, in said applying the learned error detection and correction scheme, the first memory module is further configured to use the stored check bits for the data write with no data bits set as a check bit inversion mask for data writes to and from the first memory module.

73. The system of claim 68, wherein the device driver is further executable to verify the learned error detection and correction scheme for the first memory module after said learning the error detection and correction scheme.

74. The system of claim 65, wherein the device driver is further executable to identify interleave settings for the first memory module after said identifying the first memory module as a device-enhanced memory module, wherein the interleave settings include an interleave value and a number of cache lines used for interleaving memory.

75. The system of claim 74, wherein, in said identifying interleave settings for the first memory module, the device driver is further executable to:

write a plurality of data values to each of a plurality of burst lines in a memory region of the first memory module, wherein, if the first memory module is interleaved with one or more other memory modules, a first portion of the plurality of data values are written to the memory region of the first memory module and a second portion of the plurality of data values are written to the one or more other memory modules;

read data values written to the memory region during said writing a plurality of data values; and

examine the data values read from the memory region to determine the interleave characteristics of the first memory module.

76. The system of claim 74, wherein the device driver is further executable to verify the identified interleave settings for the first memory module.

77. The system of claim 65, wherein, after said identifying the first memory module as a device-enhanced memory module, the device driver is further executable to:

read one or more hardware identification values from the first memory module; and

store the one or more hardware identification values in a memory region accessible by the device driver.

78. The system of claim 65, wherein, after said identifying the first memory module as a device-enhanced memory module, the device driver is further executable to:

write one or more configuration values to one or more of a plurality of registers on the first memory module, wherein the configuration values are configured for use during operation of an embedded device on the first memory module;

wherein the configuration values include one or more of a sleep mode timer value, a restart timer value, an error timer value, and a driver version of a device driver for the embedded device on the first memory module.

79. The system of claim 65, wherein the device driver is configured to perform said writing, said reading, and said identifying during system startup.

80. The system of claim 65, wherein the device driver is further executable to:

initialize the first memory module to operate in a system memory implementation of the system after said identifying the first memory module as a device-enhanced memory module; and

wherein, after said initializing, an embedded device on the first memory module is accessible to perform one or more functions on the system.

81. The system of claim 65, wherein an embedded device on the first memory module is a compression/decompression engine.

82. A system comprising:

a plurality of memory modules comprising system memory, wherein at least one of the plurality of memory modules comprises one or more embedded devices;

a device driver executable within the system to:

write one or more data sequences including a last data sequence to each of a plurality of memory regions in the system memory;

read at least the last data sequence from each of the plurality of memory regions in the system memory;

identify one or more of the plurality of memory modules as device-enhanced memory modules in response to said reading the at least the last data sequence from each of the plurality of memory regions in the system memory; and

wherein a first memory module of the plurality of memory modules is identified as a device-enhanced memory module if the at least the last data sequence read from a memory region on the first memory module is different than the at least the last data sequence as written to the memory region on the first memory module.

83. The system of claim 82, wherein a second memory module is identified as not being a device-enhanced memory module if the at least the last data sequence read from a memory region on the second memory module is the same as the at least the last data sequence as written to the memory region on the second memory module.

84. The system of claim 82, wherein, after said identifying the one or more memory modules as device-enhanced memory modules, the device driver is further executable to:

write a plurality of data writes to each of the one or more memory modules identified as device-enhanced memory modules, wherein each of the one or more memory modules identified as device-enhanced memory modules is configured to learn an error detection and correction scheme of the system from the plurality of data writes;

configure error detection and correction logic of each of the one or more memory modules identified as device-enhanced memory modules in accordance with said learned error detection and correction scheme;

wherein the configured error detection and correction logic of each of the one or more memory modules identified as device-enhanced memory modules is operable to provide error detection and correction for data writes to and from the memory module in accordance with the learned error detection and correction scheme.

85. The system of claim 82, wherein, after said identifying the one or more memory modules as device-enhanced memory modules, the device driver is further executable to:

identify interleave settings for each of the one or more memory modules identified as device-enhanced memory modules, wherein the interleave settings include an interleave value and a number of cache lines used for interleaving the system memory.

86. The system of claim 82, wherein, after said identifying the one or more memory modules as device-enhanced memory modules, the device driver is further executable to:

read one or more hardware identification values from each of the one or more memory modules identified as device-enhanced memory modules; and

store the one or more hardware identification values in one or more memory regions accessible by the device driver.

87. The system of claim 82, wherein, after said identifying the one or more memory modules as device-enhanced memory modules, the device driver is further executable to:

for each of the one or more memory modules identified as device-enhanced memory modules:

write one or more configuration values to one or more of a plurality of registers on the memory module, wherein the configuration values are configured for use during operation of an embedded device on the memory module;

wherein the configuration values include one or more of a sleep mode timer value, a restart timer value, an error timer value, and a driver version of a device driver for the embedded device on the memory module.

88. The system of claim 82, wherein the device driver is configured to perform said writing, said reading, and said identifying during system startup.

89. The system of claim 82, wherein the device driver is further executable to:

initialize the one or more memory modules to operate in a system memory implementation of the system after said identifying the one or more memory modules as device-enhanced memory modules; and

wherein, after said initializing, each of one or more embedded devices on each of the one or more memory modules is accessible to perform one or more functions on the system.

90. The method of claim 82, wherein one or more embedded devices on the one or more memory modules identified as device-enhanced memory modules are compression/decompression engines.

91. A memory master comprising:

a memory bus interface configured to transmit and receive data for the memory master on a memory bus; and

error detection and correction logic;

wherein the memory master is configured to receive a plurality of data writes on the memory bus interface; and

wherein the error detection and correction logic is configured to:

learn an error detection and correction scheme from the plurality of data writes received on the memory bus interface; and

apply the learned error detection and correction scheme to data including data writes to be transmitted from the memory master and data writes received by the memory master through the memory bus interface.

92. The memory master of claim 91,

wherein the memory bus interface comprises a plurality of data bits and one or more check bits;

wherein each data write of the plurality of data writes includes a data portion comprising one data bit for each memory bus data bit and a check portion comprising one check bit for each memory bus check bit; and

wherein the plurality of data writes includes one data write for each data bit in the data portion with the particular data bit set in the data portion of the data write and all other data bits in the data portion not set, and wherein zero or more of the check bits in the check portion of the data write are set to indicate an error detection and correction code of the error detection and correction scheme to be learned for the set data bit of the data write.

93. The memory master of claim 92, wherein in said learning the error detection and correction scheme, the error detection and correction logic is further configured to:

for each of the plurality of data writes with one data bit set, store the position of the set data bit at a location determined by the error detection and correction code of the data write;

wherein, in said applying the learned error detection and correction scheme, the error detection and correction logic is further configured to retrieve one or more of the stored data bit positions using the error detection and correction code associated with the particular data bit position.

94. The memory master of claim 93, wherein, in said applying the learned error detection and correction scheme to a received first data write, the error detection and correction logic is further configured to determine which, if any, of the data bits in the first data write are in error using the stored data bit positions retrievable using the error detection and correction codes associated with the data bit positions.

95. The memory master of claim 92, wherein, in said learning the error detection and correction scheme, the error detection and correction logic is further configured to:

generate check bit masks for each of the check bits in the check portion using the check portion and the data portion of each of the plurality of data writes for the plurality of data bits;

wherein, in said applying the learned error detection and correction scheme to the data, the error detection and correction logic is further configured to generate check bits for the data using the generated check bit masks.

96. The memory master of claim 92, wherein the plurality of data writes further includes one data write with no data bits of the data portion set, and wherein zero or more of the check bits for the data write are set to indicate an error detection and correction code of the error detection and correction scheme for the data portion with no data bits set, and wherein, in said learning the error detection and correction scheme, the error detection and correction logic is further configured to:

store the error detection and correction code for the data write;

wherein the stored error detection and correction code for the data write is configured for use as a check bit inversion mask in said applying the learned error detection and correction scheme.

97. The memory master of claim 91, wherein the error detection and correction logic is further configured to:

apply the learned error detection and correction scheme to a first data write received on the memory master, wherein the first data write comprises a plurality of data bits;

wherein, in said applying the learned error detection and correction scheme to the first data write, the error detection and correction logic is further configured to:

generate check bits for the first data write using the learned error detection and correction scheme;

generate a system check bit from each generated check bit; and

concatenate each of the generated system check bits with the data bits from the first data write for transmitting to another memory master.

98. The memory master of claim 91, wherein, in said learning the error detection and correction scheme, the error detection and correction logic is further configured to learn a check bit mask for each of one or more check bits used in the error detection and correction scheme;

wherein the error detection and correction logic is further configured to apply the learned error detection and correction scheme to a first data write received on the memory master, wherein the first data write comprises a plurality of data bits, wherein, in said applying the learned error detection and correction scheme to the first data write, the error detection and correction logic is further configured to:

for each of the learned check bit masks:

select one or more data bits from the first data write using the check bit mask; and

generate a check bit associated with the check bit mask from the selected one or more data bits; and

wherein the error detection and correction logic is further configured to verify the correctness of the data bits from the first data write using the generated check bits.

99. The memory master of claim 91, wherein the memory bus interface comprises a plurality of data bits, and wherein, in said learning the error detection and correction scheme, the error detection and correction logic is further configured to:

learn which one of a plurality of error detection and correction codes is associated with each of the memory bus data bit positions; and

store information on the learned associations in a syndrome array;

wherein, during said applying the learned error detection and correction scheme, the error detection and correction logic is further configured to retrieve each data bit position from the syndrome array using the error detection and correction code associated with the particular data bit position.

100. The memory master of claim 99, wherein the error detection and correction logic is further configured to apply the learned error detection and correction scheme to a first data write received on the memory master, wherein, in said applying the learned error detection and correction scheme to the first data write, the error detection and correction logic is further configured to:

generate check bits for the first data write using the learned error detection and correction scheme;

detect one or more data bit errors, if any, in the first data write using the generated check bits; and

if the detected one or more data bit errors are correctible using the learned error detection and correction scheme, correct the one or more detected data bit errors using the syndrome array to locate the one or more bits that are in error.

101. The memory master of claim 91, wherein the memory master is a device-enhanced memory module comprising one or more embedded devices.

102. The memory master of claim 101, wherein at least one of the one or more embedded devices on the memory master is a compression/decompression engine.

103. A memory module comprising:

error detection and correction scheme learning logic configured to learn an error detection and correction scheme;

check bit generation logic configured to generate check bits using the learned error detection and correction scheme for data received by the memory module; and

error detection and correction logic configured to:

check the received data using the generated check bits; and

correct the received data if an error or errors is detected and the learned error detection and correction scheme provides for the correction.

104. The memory module of claim 103, wherein the check bit generation logic is further configured to generate check bits for data to be transmitted from the memory module.

105. The memory module of claim 103, wherein the memory module is a device-enhanced memory module comprising one or more embedded devices, wherein at least one of the one or more embedded devices on the memory module is a compression/decompression engine.

106. A carrier medium comprising program instructions for detecting device-enhanced memory modules in a system including one or more memory modules comprising system memory, wherein at least one of the one or more memory modules is a device-enhanced memory module comprising one or more embedded devices, wherein the program instructions are computer-executable to implement:

writing a data sequence to a first memory module of the one or more memory modules in the system;

reading the data sequence from the first memory module;

identifying the first memory module as a device-enhanced memory module if the data sequence as read from the first memory module is modified from the data sequence as written to the first memory module; and

identifying the first memory module as not being a device-enhanced memory module if the data sequence read from the first memory module is the same as the data sequence as written to the first memory module.

107. The carrier medium of claim 106, wherein the program instructions are further computer-executable to implement:

after said identifying the first memory module as a device-enhanced memory module:

writing a plurality of data writes to the first memory module;

learning an error detection and correction scheme of the system from the plurality of data writes; and

configuring error detection and correction logic of the first memory module in accordance with said learned error detection and correction scheme.

wherein the configured error detection and correction logic of the first memory module is operable to provide error detection and correction for data writes to and from the first memory module using the learned error detection and correction scheme.

108. A carrier medium comprising program instructions for learning an error detection and correction scheme in a system including one or more memory modules comprising system memory, wherein at least one of the one or more memory modules is a device-enhanced memory module comprising one or more embedded devices, wherein the program instructions are computer-executable to implement:

writing a plurality of data writes to a first memory module of the one or more memory modules, wherein the first memory module is a device-enhanced memory module;

learning the error detection and correction scheme of the system from the plurality of data writes; and

configuring error detection and correction logic of the first memory module in accordance with said learned error detection and correction scheme;

wherein the configured error detection and correction logic of the first memory module is operable to provide error detection and correction for data writes to and from the first memory module using the learned error detection and correction scheme.

5